A Parallel Infrastructure and Programming Model for Adaptive, Irregular Applications

S. Lang[a]

[a]Interdisciplinary Center for Scientific Computing
University of Heidelberg
Heidelberg, Germany 69120

Parallel applications based on the message passing paradigm are difficult to design and implement. The development process sharpens when furthermore solution adaptive techniques are needed and computations on complicated 3D-geometries are focused.
Many the arising difficulities are addressed inside the presented infrastructure 'UG' (Unstructured Grids), for the computation of pde's, e.g. dynamic load migration and load balancing, parallel local grid adaption, and parallel IO.
We present proper abstractions for each of the different functionality parts of a parallel, adaptive and unstructured software system. This architecture assures a maximal degree of code reuse. Therefore the treatment of various partial differential equations is possible without superfluous coding.
The parallel grid manager handles a hierarchical multigrid data structure, describing a two- or threedimensional geometry. On top of this manager works the parallel and local adaption process, which controls manipulation of the distributed meshes. It allows refinement and/or coarsening of grids consisting of mixed element types. Each adaption pattern results in a grid closure such that no hanging nodes occur and the (multi)grid is always kept consistent. In our implementation grid adaption shows a truly local behaviour, thus it is well suited for parallelisation.
In a parallel context grid adaption involves the need to rebalance the computational load. This stage involves both determining a new load balancing and dynamically redistributing the objects of the grid parts. We discuss load balancing for multigrids in general and present methods based on multiple constraints, which are able to dynamically rebalance the multigrid.
A key feature of 'UG' is the capability to dynamically migrate grid objects between the processors during run time. Thus the computation need not be interrupted, but continues after load transfer with a balanced work load on each processor. This difficult task is supported by 'DDD' (dynamic distributed data), a new parallel programming model. The migration process with its various stages is discussed: packing the data objects into buffers, sending and receiving the message buffers and unpacking the data objects.
The DDD implementation contains several components with different functionality,

- Interfaces, to supports communication operations efficiently,
- Transfer, providing procedures for creating and deleting distributed object copies,
- Identification/Join, to create one global logical object out of several, and processor local objects
- Prio for changing priorities in a global consistent manner.

Finally we present parallel adaptive calculations of real-world problems. Timings, efficiency and scalabilty of parallel grid adaption, load balancing and load migration are compared with the numerical phases, discretization and solution. The results clearly show the enourmous possiblities when the above acceleration strategies - multigrid, adaptivity and parallelism - are combined.

References

[1] UG, http://cox.iwr.uni-heidelberg.de/∼ug